UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/655,596	09/06/2000	William F. Beausoleil	POU9-2000-0048-US1	9320

34313 7590 01/03/2005

ORRICK, HERRINGTON & SUTCLIFFE, LLP
4 PARK PLAZA
SUITE 1600
IRVINE, CA 92614-2558

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 01/03/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/655,596	BEAUSOLEIL ET AL.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication app ars on th cover sheet with the correspond nce address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 August 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 8/24/2004.

As indicated in Applicant's response, no claims have been amended. Claims 1-4 are pending in the office action.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beausoliel et al., USPN: 5,551,013 (hereinafter Beausoliel) in view of Austin et al., USPN: 4,885,684 (hereinafter Austin); and further in view of Baker et al, USPN: 5,701,502 (hereinafter Baker).

As per claim 1, Beausoliel discloses a emulation engine comprised of a plurality of modules, a work station, and a bus for transferring data between the work station and said modules (Fig. 1,8), each of such modules including a plurality of processors and a module main memory accessible for data transfers during an emulation by each of such processors (e.g. col. 3, line 55-65; col. 5, lines 32-35; Fig. 3A-B), each of such processors having a control store to store a programmable sequence of emulation steps that define logic states for its processor (e.g. col. 4, lines 1-5; col. 6, lines 2-10; Fig. 9A, 11A), a method to allow data transfers between such module main memory unit and work station without interrupting an in-progress emulation (*non-blocking* - col. 8, lines 16-56; Fig. 5,7), comprising:

compiling said programmable sequence of emulation steps to include, in at least one step, a blocking code, when the step is read from the control store (e.g. col. 10, line 64 to col. 11, line 29; *Control Store* - col. 5, line 39 to col. 6, line 18; Figs. 2-3), as a disable command between the plurality of said processors and main memory unit (e.g. *active, inactive* - col. 6, lines 14-27, 28-59);

decoding said blocking code (e.g. col. 12, Table 1- Note: decoding is inherent to encoding instructions; Fig. 2a-b; Fig. 3a-b – Note: control word field/bit extraction is equivalent to decoding instructions code); and

transferring data between said work station and module main memory (e.g. *input to target system, emulation support facility*- col. 3, lines 28-37 ; *workstation* - Fig. 8 – Note: workstation is equivalent to host computer coordinating the support facilities functions operable on the processor modules, e.g. data or instructions transfer)

But Beausoliel does not specify a maintenance bus for transferring data between the workstation and processor modules. However, Beausoliel discloses latches and multiplexers to control data flow into or out of the execution needed by the emulation logic; and path control bits multiplexing and changes for allowing concurrent data connection to support facilities in the emulation environment (e.g. col. 3, lines 28-37; col. 8, lines 16-56; Fig. 1, 2A, 3A-B). Austin, in a network of processing units simulation environment with emulation, programmability/debug and interprocessor communication/scheduling and management support (see cols. 4-13; col. 17, lines 22-46) analogous to that of the emulating network of Beausoliel such as to use software program for deploying/supporting the functionality of the distributed data processing, discloses a maintenance bus for both the processing elements and for the supervisor unit (e.g. *EMB* and *TM*

bus - col. 6, lines 4-26) for update storage operations, and other off-line functions. In view of the suggested teachings from Beausoliel's and Austin's from above, it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide a maintenance bus as taught by Austin to Beausoliel's concurrent support facilities operable within the emulation execution and operations by the processing units of Beausoliel's system because this would provide a specialized communicating channel, e.g. bus, to support the maintenance operations as suggested by Austin without affecting or interrupting the main data flow used by the processing units of the emulation by Beausoliel, thereby obviate bus/memory contention issues and enhance fault prevention, efficiency.

Nor does Beausoliel explicitly specify that in response to decoding blocking code, blocking transfers between the plurality of module processors and said module main memory units; and transferring data between the workstation and module memory units during such blocking from above. But the disable command from decoding the MOP bit teaches disabling access by the processors to specific parts of memory. This implicitly discloses a blocking of transfer between the processors and the memory. But in case Beausoliel does already not provide such blocking of transfer as a result of a status, modifying a disable code to a blocking code would have been obvious.

Austin, in the system for supporting and deploying distributed processors via software as mentioned above, discloses the use of maintenance bus to support the operations of the processing modules (re col. 6, lines 4-26); and teaches, via the use of LOS (Local Operating System), initialization, debug and error-related suspension operations within the emulating unit processors and/or recovery operations using such maintenance bus (e.g. col. 7, lines 51-63;

suspension, minimal overhead - col. 12, line 47 to col. 13, line 7; *maintenance bus, alternate paths* - col. 9, lines 3-33, 42-50). The use of maintenance facilities to allow recovery of faulty modules and to prevent contention of memory access during such recovery or maintenance tasks or the use of a protocol implemented via control bits to reject or block memory access to bus-connected processors or requests was a known concept as evidenced herein with Baker. Baker, also in an environment to use software or microcode to integrate functions of a target system emulating processors based on a existing processor operating system to achieve fault-tolerant system analogous to the integrated development network using code simulation by Baker, discloses a maintenance bus and halting of process communications between the faulty processing unit and the central operating system in response to decoding a request to handle a maintenance interrupt (e.g. *lock-step* -col. 119, line 4-53) and thereby invalidating of non-valid data transfer to memory (e.g. col. 127, lines 3-23). Hence, in view of above-mentioned Beausoliel's teachings on support facilities and Austin's use of maintenance bus in conjunction with Baker's scheme to suspend servicing of slave processing units memory requests for synchronization tasks as a result of a maintenance interrupt detection as mentioned above, it would have been obvious for one of ordinary skill in the art at the time the invention was made to implement code instructions as taught by Beausoliel so that when a blocking code is decoded, the transferring of data between processors and their memory units is blocked so that the maintenance bus is utilized in order to allow data be transferred from the maintenance dedicated workstation onto the main memory unit just as suggested by Austin and by the recovery synchronization calls by Baker for the same reasons as mentioned above in the rejection of the maintenance bus limitation and also because this would avoid further contamination/incoherency

of the processor memory when external data are written to their memory units, i.e. synchronization process as taught by Baker.

As per claim 2, Beausoliel does not specify the step of unblocking transfers between the module processors memory unit when such step is sequentially decoded next after a blocking code step. But in view of the combined teachings by Austin and Baker's teachings on suspending operations upon a failure detection (Austin: col. 7, lines 51-63; *suspension, minimal overhead* - col. 12, line 47 to col. 13, line 7; *maintenance bus, alternate paths* - col. 9, lines 3-33, 42-50; Baker: lock-step - col. 119, lines 12-24) as mentioned above, it would have also been obvious to add the step of unblocking after a decoded step of blocking has been completed to Beausoliel's repetitive emulation process because the motivation would be that once the maintenance steps as suggested by Austin are completed, it would be necessary to resume the data transfer and communication between the processors and their memory unit or bus system in Beausoliel's system in order to proceed on with the rest of the emulation.

As per claim 3, Beausoliel discloses a repetitive cycle in decoding and emulating the program code (e.g. col. 12, lines 5-13) but fails to specify that the transferring of data step between module memory and workstation is repetitive. But in view of the rejection of such data transferring step as addressed in claim 1, the repetition of such data transfer would be inferred from the combined teachings by Beausoliel (repetitive emulation decoding) and Austin combined with Baker (maintenance bus, LOS, suspension and initialization operations) and the rejection as set forth above.

As per claim 4, Beausoliel discloses a repetitive cycle in decoding and emulating the program code (e.g. col. 12, lines 5-13) but fails to specify that the transferring of data step

between modules memory units and workstation being followed by an unblocking step is repetitive. This limitation corresponds to the same limitation of claim 3 above; hence, in view of the combined teachings by Austin/Beausoliel/Baker and the rejection as set forth in claims 2, and 3 above, this limitation would have been obvious because of the rationale as set forth therein.

Response to Arguments

4. Applicant's arguments with respect to claims 1-4 have been considered are not persuasive and in that respect, deserve some counter arguments.

(A) Applicants have submitted that Beausoliel 'does not teach or suggest a blocking code ... blocking transfers ... plurality of module processors ... main memory' (Appl. Rmks, pg. 5, last para); that MOP bit in the control word is "active" ... whether the processors will emulate a logic function ... during execution of that particular word ... MOP bit is not a "blocking code" that causes ... main memory'; and that Beausoliel only discloses a series of control words (Appl. Rmks, pg. 6, 1st , 2nd para). The claim recites "compiling ... emulation steps to include ... one step, a blocking code" that when decoded from the control store, the step is decoded as a disable command. Hence, as construed from reading the claim, a control store with stored therein a instruction or data structure being a component from the emulation program, such data being decoded as one step and become a disable command between the main memory and the processors communicating with the memory. In col. 10, line 64 to col. 11, line 2, Beausoliel's emulation program includes storing of a component, in form of words, in a control store to support control on the emulation program; and this control store with the control words described in Figs 2-3 amounts to decoding steps leading to a command disabling or enabling memory access in one way or another. Beausoliel's disabling command reads on denying if not blocking

access to certain memory part that would otherwise be accessible had the control word in the control store not have a specific status as evidenced in the cited parts of the rejection in regard to the MOP bit. The claim further recites 'decoding said blocking code and ... blocking transfers between the plurality of ... processors and ... main memory'. The term 'blocking code' is not specific enough, hence has been construed as a form of artificial information that upon being decoded dictates some action such as to disable or prohibit access of part or all of resources, e.g. memory for read or write; and this is met by the MOP bit going one way or the other as cited. The claim does not describe in more precise terms how the so-called blocking code would clearly distinguish over what has been interpreted by Examiner and used in the rejection: a bit whose status dictates that some part of memory cannot be accessed reads on access disabling or blocking code, e.g. if a part of a memory location is prohibited for access (see col. 6, lines 52-59), what transfers there are between that part of memory and a plurality of processors being involved would be disabled, if not blocked. There is no specificity in the claim as far as how completely or partially this blocking is to be effected; hence, if transfer like read/write between the emulated processors and any part of memory such as shown in the reference; and this disabling command is construed as a form similar to blocking. In case such blocking code is not explicit in Beausoliel, this limitation would have been obvious as set forth in the rejection.

(B) Applicants have submitted that Austin does not teach a 'blocking code ... decoding said blocking code and in response thereto ... said module main memory' (Appl. Rmrks, pg. 6, 3rd para). The rejection is geared to show that the use of maintenance bus would have been obvious in emulation or integrated simulation of network of interconnected processors comprising a main memory. The argument about 'blocking code' being decoded and as a result thereof transfers are

blocked has been addressed in section A above. The Applicants fail to point how the rationale as to why a maintenance bus being added to Beausoliel's system would be inappropriate, and if so for what specific reasons. Hence, the arguments amount to mere allegations.

(C) Applicants have submitted that Baker does not teach a 'blocking code ... decoding said blocking code and in response thereto ... said module main memory' (Appl. Rmrks, pg. 6, bottom para); and that, as combined, neither Beausoliel, Austin, or Baker teaches the blocking code as required in claim 1 (Appl. Rmrks, pg. 7, 1st para). Baker is used to emphasize the use of maintenance bus (taught by Austin) and to inhibit unwanted transfers between processors and the main memory of a interconnected processing system comprising operation analysis and control management. The concept of transfer blocking to allow the use of maintenance bus and corrective, synchronization functions as well as update operations is at issue here when the teachings by Baker in conjunction with the maintenance bus by Austin, to provide a blocking step just for that purpose, the concept of control bit to disable memory access already taught by Beausoliel. The Applicants fail to show how the teachings by Baker, in conjunction with the memory disabling and the bus maintenance by Austin would teach away from each other references or would yield an unwanted outcome with respect to Beausoliel, Austin or Baker's method/invention.

(D) Applicants have submitted that there is no motivation to combine because of unrelated technologies (Appl. Rmrks, pg. 7, last para, pg. 8, top). As a whole, each reference (i.e. Beausoliel, Austin or Baker's method/invention) involves a central system including memory and control means to manage and check operation of interconnected processors; and the point at issue is to use a maintenance paradigm using a bus and some form of code to allow the normal

operations to be stopped between said memory and plurality of processing units being monitored or dispatched or observed by a central management logic. The rejection has pointed out the reasons why for combining disabling of memory access from Beausoliel, a maintenance bus and stopping interactions of unwanted processors data with memory, e.g. needs for memory integrity or applying of corrective actions to the memory resource. The fact that the environment under which interconnected processors in the references slightly differ from each other does not negate a useful methodology whose teaching is at stakes here; that is, if a interconnected multi-processor system operating using a common bus and memory, and provision for disabling a common memory such as cited in Beausoliel and Austin and Baker's attempt to keep valid data in memory by stopping unwanted transfers between processors and a common memory. Maintenance of memory so that only wanted and valid data are kept is the methodology behind having blocking code and a maintenance bus. Applicants fail to provide why such teaching is not appropriate when one skill in the art has under his disposition the teachings from Beausoliel, Austin, and Baker.

In short, Applicants' argument about references not being analogous is not persuasive. That all the references presented should come from analogous fields of endeavor is not a strict requirement according to current rules and procedures governing USC 103(a) rejection. It is an application and/or useful methodology that needs to be addressed not the environment wherein such methodology is being suggested or disclosed. Hence, by applying a motivation when combining the teachings from 3 references as set forth in the rejection, a prima facie case has been established; and the mere allegation that the references used come from 3 technologies does not amount to overcome the rationale as set forth in the rejection.

The claims will stand as set forth in the rejection.

Conclusion

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (272) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571)272-3719.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence – please consult Examiner before using) or 703-872-9306 (for official correspondence) or redirected to customer service at 571-272-3609.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

VAT
December 19, 2004

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100